U.S. Department
of Transportation

United States
Coast Guard

Commandant
United States Coast Guard

2100 Second Street, S.W.
Washington, DC 20593-0001
Staff Symbol: G-SIA
Phone: 202-267-6857
DEC 15, 1999

COMDTINST 5230.59
DEC 15, 1999

COMMANDANT INSTRUCTION 5230.59

Subj: U.S. COAST GUARD COMMON OPERATING ENVIRONMENT (USCG COE)

Ref: (a) Information Technology Management Strategy, June, 1998
(b) Information Systems Technical Architecture (ISTA), COMDTINST M5230.45 (series)
(c) Coast Guard Software Development and Documentation Standards (CG-SDDS), COMDTINST 5234.4
(d) Standard Workstation III Configuration Management Policy , COMDTINST 5200.16
(e) Desktop Productivity Software Standards, COMDTINST 5234.5
(f) Standard Workstation Operating System Standards, COMDTINST 5230.2
(g) Coast Guard Data Element Naming Standards, COMDTINST 5230.42 (series)
(h) USCG C4I Baseline Architecture, COMDTINST 3090.6
(i) C4I Objective Architecture and Transition Plan, COMDTINST 3090.7

1. PURPOSE. This directive promulgates the U.S. Coast Guard Common Operating Environment (USCG COE) and declares the policy and guidance for COE compliance in the development of all U.S. Coast Guard Information Technology (IT) initiatives.

2. ACTION. Area and district commanders, commanders of maintenance and logistics commands, commanding officers of headquarters units, assistant commandants for directorates, chiefs of offices and special staff divisions at headquarters shall ensure compliance with the provisions of this directive.

3. DIRECTIVES AFFECTED. None.

4. BACKGROUND. The Office of Architecture and Planning (G-SIA) is responsible for the U.S. Coast Guard's Information Architecture (IA). The USCG COE is one of the components of the IA. Reference (a) describes the overarching strategy for managing IT. The concepts described provide a foundation from which the USCG COE is built. Reference (b) outlines specific standards upon which the U.S. Coast Guard IA is built. Reference (c) defines software development and documentation standards for automated information systems, and assigns Commandant (G-SIA) the oversight authority for ensuring U.S. Coast Guard development initiatives comply with the CG-SDDS. Reference (d) establishes policy COMDTINST 5230.59 and the

process for Configuration Management (CM) of the CG desktop workstation environment. This CM process provides an entry point for managing changes to the USCG COE. References (e) and (f) describe specific functional elements of the USCG COE that provide the foundation for the U.S. Coast Guard's desktop workstation environment. These documents also specify the primary end-user interface to which developers must write their applications. Reference (g) delineates the U.S. Coast Guard Data Element naming standards and policy on which the CG Data Architecture is built. Reference (h) defines the current state C4I and Sensors Architecture and C4I Operational Requirements. Reference (i) contains the U.S. Coast Guard's vision for its future C4I Architecture, the Objective Architecture, and how to make that vision a reality, The Transition Plan.

5. DISCUSSION. The Government Performance and Results Act (GPRA) and the Clinger-Cohen Act require the Federal government to implement improvements to its management of IT, measure the outcomes, and report the results of those improvements. The U.S. Coast Guard has recognized that certain functional elements of its information infrastructure are so fundamental, that they must be considered in virtually all Enterprise Applications (EAs). The Information Technology Management Board (ITMB) chartered the USCG COE Quality Action Team (QAT) in October 1997 to define the components, and related policy of the USCG COE. Enclosure (1) is the product of that effort. The USCG COE provides a framework for the standards, policy, and guidance to be applied throughout the entire lifecycle of all U.S. Coast Guard IT initiatives. The Department of Defense (DoD) maintains the Defense Information Infrastructure Common Operating Environment (DII COE). The USCG COE has been modeled after the (DII COE) to ensure interoperability with DoD. Compliance with the USCG COE to leverage technology uses scarce resources more effectively. The USCG COE and referenced documents should be recognized as living documents, and as such, frequent updates can and should be expected. The USCG COE is available on the U.S. Coast Guard's Intranet at: cgweb.uscg.mil/g-s/g-si/g-sia/ita/cgcoe/index.htm.

6. POLICY. The USCG COE is the catalog of mandatory common IT products for meeting all U.S. Coast Guard IT requirements. All U.S. Coast Guard Information Technology solutions being acquired, developed, or implemented in response to CG information technology requirements shall comply with all provisions defined in the USCG COE and related directives. Any IT initiatives proposed as solutions to requirements not contained in enclosure (1), shall submit a copy of a completed SWIII Engineering Change Proposal (Reference (d), enclosure 2) directly to Commandant (G-SIA), for processing through the USCG COE configuration management process, as described in enclosure (1). The CIO is the sole authority for approving any changes to the USCG COE.

7. POINT OF CONTACT. Direct all questions or comments relating to the USCG COE to the Headquarters' Office of Architecture and Planning, G-SIA, cgcoe@comdt.uscg.mil, 202-267-6857.

GEORGE NACCARA
Director of Information and Technology

Encl: (1) U.S. Coast Guard Common Operating Environment (CG COE)

ENCLOSURE (1) TO COMDTINST 5230.59

U.S. Coast Guard
Common Operating Environment
(CG COE)
DECEMBER 1999

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY 1

2. CG COE LAYERS DEFINED 1
2.1. Technical Architecture Layer 2
2.2. Kernel Layer 2
2.3. Infrastructure Services Layer 2
2.4. Common Support Applications Layer 3
2.5. Applications Programmer Interfaces (API) Layer 3

3. CG COE FUNCTIONAL ELEMENTS DEFINED 3
3.1. Technical Architecture Layer 4
3.2. Kernel Layer 4
3.2.1. Operating System (OS) and Windowing Services. 4
3.2.2. System Management Services. 4
3.2.3. Security Management Services. 5
3.2.4. Executive Manager. 5
3.2.5. COE Tools. 5
3.2.6. Print Services. 5
3.2.7. Network Services. 5
3.2.8. Window System Services. 5
3.3. Infrastructure Services Layer 6
3.3.1. Management Services. 6
3.3.2. Communications Services. 6
3.3.3. Distributed Computing Services. 6
3.3.4. Presentation Services. 6
3.3.5. Data Management Services. 6
3.3.6. Workflow and Global Data Management Services. 7
3.3.7. Data Interchange Services. 7
3.3.8. Electronic Data Interchange (EDI) Component. 7
3.4. Common Support Applications Layer 7
3.4.1. Messaging Processing Services. 7
3.4.2. Office Automation Services. 8
3.4.3. Data (Logistics) Analysis. 8
3.4.4. Developer's Toolkit. 8
3.4.5. MCG&I Mapping Cartography Geodesy & Imagery (MCG&I). 8
3.4.6. Alerts Service. 8
3.4.7. Correlation Service. 8
3.4.8. On-line Help Services. 9
3.4.9. Data Access Services. 9
3.5. Applications Programmer Interfaces (API) 9

4. CG COE MAINTENANCE PROCESS 10

TABLE OF CONTENTS

5. IT MANAGEMENT STRATEGY 12
5.1. Concepts 12

6. IT IMPLEMENTATION STRATEGIES 14

7. STANDARDS, OVERVIEW 14
7.1. Common Products List 15
7.2. Certified Enterprise Applications 15
7.3. Workstation/Server Architecture 15

Appendix 1 - Index of Related Policy Documents 16
Appendix 2 - Glossary 17
Appendix 3 - Workstation/Server Architecture 18

U.S. Coast Guard Common Operating Environment (USCG COE)

1. EXECUTIVE SUMMARY

The Clinger-Cohen Act of 1996 (formerly known as the Information Technology Management and Reform Act (ITMRA)), mandates agency Chief Information Officers (CIOs) to "improve the efficiency and effectiveness of agency delivery of products and services to the public through the effective use of information technology." As one of many ongoing initiatives to meet this mandate, the Coast Guard CIO chartered a Quality Action Team (QAT) to publish the first U.S. Coast Guard Common Operating Environment (USCG COE). This document is result of that effort.

The USCG COE concept encompasses all CG functional/business areas including Mission, Oversight and Control, and Support, as defined in the Coast Guard's Information Architecture. The intent of the USCG COE is to provide an underlying framework that integrates a suite of components to form an evolutionary acquisition and implementation strategy. The USCG COE emphasizes a continuous evolution of a stable Information Technology (IT) baseline through incremental development, testing, and fielding, to take advantage of new technologies as they mature. Successive releases are narrowly focused to maximize product stability. The changes are iterative so users always have a stable operating environment.

The USCG COE is a template of mandatory IT standards, policy, and procedures that facilitates quality, interoperable, portable, information systems composed of reusable components. These systems peacefully co-exist, with a reduced life cycle and cost. In other words, USCG COE compliance promotes faster delivery of better information systems that are cheaper to operate and maintain.

The intended audience of the USCG COE is all personnel involved in any phase of a Coast Guard information system's life cycle. The USCG COE provides a documented, stable, enterprise-oriented, target operating environment in which to implement Coast Guard systems. All system components are considered, including software, hardware, communications, and sensors.

This first iteration of the USCG COE defines its underlying purpose, philosophy, structure and components. Follow-on work to this effort will integrate the Coast Guard Information Systems Technology Architecture, COMDTINST 5230.45 (series), as well as the work completed by the Chief of Staff chartered C4I & Sensors Project resulting in the U.S. Coast Guard C4I Baseline, COMDTINST 3090.6 (series), and the USCG C4I Objective Architecture and Transition Plan, COMDTINST 3090.7 (series). These three documents remain authoritative sub-components of the USCG COE as this follow-on work is completed.

U.S. Coast Guard Common Operating Environment (USCG COE)

2. CG COE LAYERS DEFINED

What's not in?

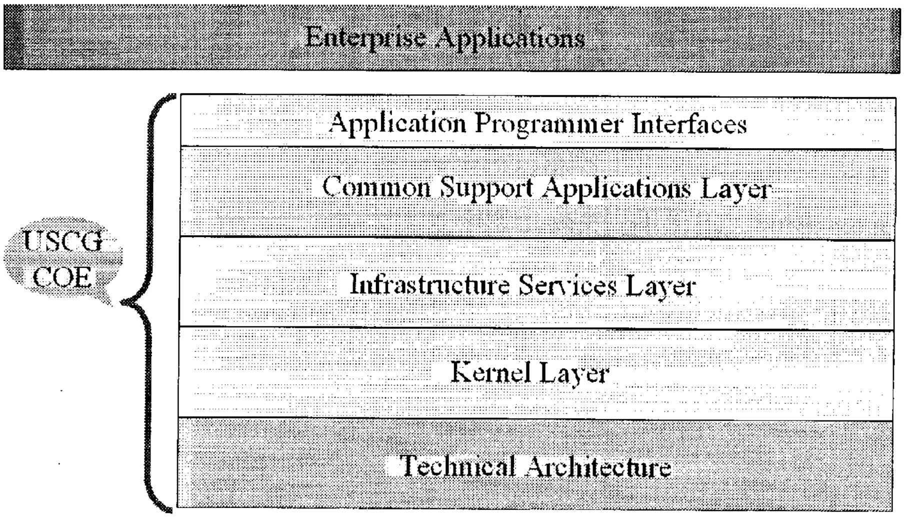
The USCG COE is not a system unto itself; it is a framework for building an open information system architecture and infrastructure. To understand the USCG COE concept better, the scope and bounds need to be defined. By establishing what is NOT in the USCG COE first, the remainder of the document can expand on the contents of the USCG COE.

Enterprise Applications (EAs) are outside the scope of the USCG COE. EAs are the information systems that run on top of the USCG COE. EAs are documented in the Coast Guard's Systems Plan. The USCG COE defines the information technology environment that EAs must operate within to ensure EA peaceful coexistence. Any new or modified EAs must be certified by TISCOM to successfully function without affecting other components within the USCG COE to be deployed.

Data Standards are outside the scope of the USCG COE. COMDTINST 5230.42 (series), U.S. Coast Guard Data Element Naming Standards, describes the Coast Guard Data Architecture and establishes provisions by which data that is processed by USCG COE compliant systems shall conform to. The objective is to design applications separate from the data that is processed to promote principles such as data sharing, source data entry, and interoperability.

What's in?

The USCG COE is a "plug and play" open architecture. Components may be added or deleted through a structured process, as required to maintain the living model. The following graphic depicts the layers of the model and is intended to establish a common vocabulary for describing the set of services and interfaces of the USCG COE. The definition for each layer follows.


Enterprise Applications
Application Programmer Interfaces
Common Support Applications Layer
USCG COE
Infrastructure Services Layer
Kernel Layer
Technical Architecture

U.S. Coast Guard Common Operating Environment (USCG COE)

2.1. Technical Architecture Layer

In reference to the previous illustration, the Technical Architecture (TA) is the foundation layer of the USCG COE. This layer defines the U.S. Coast Guard standard IT hardware including computers, communication components, electronics, and sensors that are used to acquire, store, process, and transmit information. Periodic updates to the TA, leveraging rapid advancements in technology, are necessary to address new or changing IT requirements. Commandant G-SIA manages the configuration of the TA through the USCG COE configuration management process described in section 4.0.

The USCG C4I Baseline Architecture, COMDTINST 3090.6 (series), describes the USCG C4I and sensors baseline architecture as a subset of the Technical Architecture. This document is maintained by G-OCC, the Office of Command and Control Architecture. The most current configuration may be found at the following web site: http://cgweb.uscg.mil/g-s/g-si/g-sia/ita/docs/c4i%baseline/index.htm

The U.S. Coast Guard Standard Workstation III COE describes the Coast Guard's Office Automation desktop environment as another subset of the Technical Architecture. The most current configuration may be found at the following web site: http://cgweb.tiscom.uscg.mil/isd/ENG1/ConfMgmt/cfgmgthome.htm

The Information Systems Technical Architecture (ISTA), COMDINST M5230.45 (series), describes the IT standards which support the elements of the Technical Architecture. This document is maintained by Commandant, G-SIA. The most current version may be found at the following web site: http://cgweb.uscg.mil/g-s/g-si/g-sia/ita/ista/DOCS/COMDTINST523045a.doc

2.2. Kernel Layer

The COE kernel is the mandatory minimal set of software required on every workstation to provide certain common services regardless of platform, or purpose. The base of the Kernel layer is a Portable Operating System Interface (POSIX)-compliant operating system. The USCG COE kernel includes the Operating System, Windowing Services, Graphical User Interface (GUI) and commercial product elements that provide additional functionality (e.g. system administration, security administration, executive management, administration templates, as well as USCG COE installation tools). This is the foundation of the USCG Standard Workstation standard image.

2.3. Infrastructure Services Layer

Infrastructure Services are a set of capabilities provided to support mission applications and systems operations on an as- required basis. Examples of Infrastructure Services include data management services, presentation services, and communications services. Infrastructure services expand and supplement the basic services provided in the kernel and are installed at the discretion of a site administrator.

2.4. Common Support Applications Layer

Common support applications are an upper level layer of applications software and toolkits designed to provide specific functionality to an end user or to support the development of an application or process (e.g. Office Automation services, developer's toolkit, etc.). Support applications are tailored to user functions that will be performed at a workstation. Support applications consist of COTS (commercial off-the-shelf) and GOTS (government off-the-shelf) products.

2.5. Applications Programmer Interfaces (API) Layer

APIs provide software applications access to the Platform Services or to the COE component services. A programmer invokes specific services in a software application by using API calls. Standards-based API calls (as specified in the COE) provide a level of application portability and independence from the underlying Platform Services.

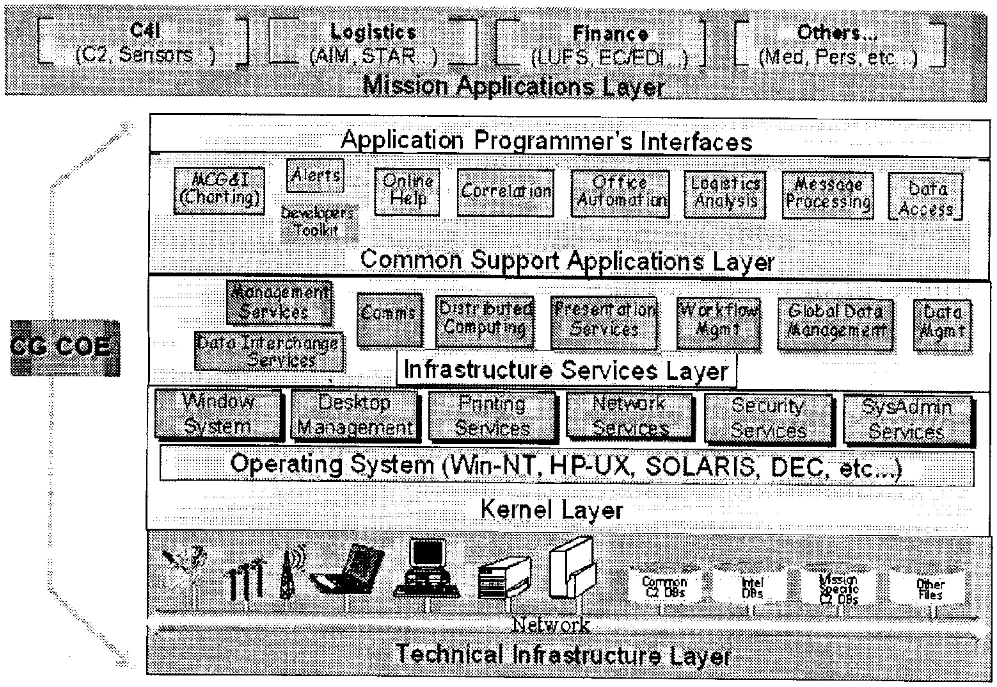
3. CG COE FUNCTIONAL ELEMENTS DEFINED

In COE-based systems, all software and data - except certain portions of the kernel such as the operating system and basic windowing software - are packaged in self-contained units called functional elements. Functional elements are the most basic building blocks from which a COE-based system can be built. Functional elements are defined in terms of the functionality they provide. A functional element may consist of one or more "modules." They are defined as a collection of related functions as seen from the perspective of the end user, not the developer. The reason for defining functional elements in this way is that it is a more natural way of expressing and communicating what software features are to be included in the system than by individual process, file name, or data table.

For example, it is more natural to think of a system as containing a Message Processing Functional element than executable software files called MP_In.exe and/or MP_Out.com. It's easier for the end user to think of a Word Processor Functional element than a software module that opens a file, another module that paginates a file, another module that compresses a file, etc.

Those functional elements that are part of the COE are known as COE-component functional elements, or more precisely, as functional elements that are contained within the COE based on their functionality. Functional elements that are built on top of the COE to provide capabilities specific to a particular mission domain are mission-application functional elements and exist in the Mission Applications Layer. The principles which govern how functional elements are loaded, removed, or interact with one another are the same for all functional elements, but COE-component functional elements are treated more strictly because they are the foundation on which the entire system rests. The following USCG COE model illustrates this modular concept:

U.S. Coast Guard Common Operating Environment (USCG COE)



3.1. Technical Architecture Layer

This layer consists of the components that make up the Technical Architecture. This includes file servers, database servers, peripherals, and the network and related hardware that connects the computers. A more detailed description is found in the Standard Workstation Architecture, and the C4I and Sensors Baseline Architecture documents.

3.2. Kernel Layer

Operating System.
A COE kernel will always contain the operating system and windowing environment, but it will also include the following features:

3.2.1. Operating System (OS) and Windowing Services.

The operating system provides a standard release of a vendor-specific OS and a specified set of patches that must be applied to guarantee a standard runtime environment and well-behaved execution of layered COE variants.

3.2.2. System Management Services.

The System Management Services functional element contains the software necessary to perform basic system administration tasks such as:

- reboot or shut down the system
- user account and profile management
- format hard drives
- load or install products and services

3.2.3. Security Management Services.

The Security Management Services functional element is used by the security administrator to enforce system security policy. The operating system and other COE components provide security policy enforcement. Functional elements loaded later may provide additional system and security administration capabilities, but the minimum capabilities for security enforcement and security administration are in the kernel.

3.2.4. Executive Manager.

The Executive Manager component of the kernel is the interface through which an operator issues commands to the system. The Executive Manager is an icon-and-menu-driven desktop interface, not a command-line interface. The templates included in the COE kernel are used to define the basic runtime environment context that an operator inherits when they log in; which processes to run in the background, and which environment variables are defined.

3.2.5. COE Tools.

The COE tools within the kernel allow other functional elements to be installed and enforce critical COE principles. The COE kernel assures that every platform in the system operates and behaves in a consistent manner and that every platform begins with the same environment.

3.2.6. Print Services.

Print Services provide the basic print capability of the system. They provide such functions as user selection of a default printer, printer administration, and a common way of accessing print resources from an application program and remote printer administration.

3.2.7. Network Services.

The Network Services functional element provides Domain Name Services (DNS).

3.2.8. Window System Services

Microsoft Windows NT(r). Microsoft Windows NT provides the intrinsic graphical user interface, incorporating the WIN32 user-interface standard. This is the standard windowing interface for the CG Windows NT platforms.

HP UNIX.

X Windows. Massachusetts Institute of Technology (MIT)'s X Windows X11R5(r) provides a network-transparent communication protocol between an application and its presentation logic, high-performance device- independent graphics, and a hierarchy of resizable, overlapping windows. This is the standard windowing package for CG Unix-based platforms.

MOTIF. The Open Group's MOTIF(r) Window Manager is the graphical user interface built on X Windows. This is the standard windowing interface for USCG platforms running the UNIX operating system.

With the Coast Guard in a Windows NT environment, procurements should stipulate the WIN32 standard wherever practicable. However, if interoperability is needed with an agency or organization that uses X-Windows, the procurement must consider the X-Windows standard. FIPS PUB 158-1 addresses layers 0, 1, and 2 of the User Interface System Reference Model (UISRM) which is based on the Massachusetts Institute of Technology's X Window system. Work is still in progress on Layers 3, 4, 5, and 6 and has not progressed enough to include specifications in this model on those layers.

Menus - Allows for the generation of custom menus within the desktop environment and the customization of existing applications in the desktop.

Icons - Allows for the creation of pictorial representations of applications associated with the information which the operating system needs to execute the application.

3.3. Infrastructure Services Layer

Infrastructure Services are largely independent of any particular application.

3.3.1. Management Services.

The Management Services functional element includes network and system management, and security administration.

3.3.2. Communications Services.

The Communications Services functional element provides facilities for receiving and sending data out of the system to interact with other message and data systems.

3.3.3. Distributed Computing Services.

The Distributed Computing Services functional element provides the infrastructure necessary to achieve true distributed processing in a client/server environment.

3.3.4. Presentation Services.

The Presentation Services functional element is responsible for direct interaction with the human whether that is through windows, icons, menus, or multimedia.

3.3.5. Data Management Services.

The Data Management Services functional element includes an application-independent capability to maintain and administer data through a COTS relational database management system. It also provides the user with file management in a distributed environment through a graphical interface to the file system, including "drag-and-drop" functionality between objects and between cooperating client applications, and a general file type association database. USCG COE DBMS products must support data spread across multiple sites. Data may be replicated or fragmented to improve responsiveness and increase availability. The objective is to provide location transparency, meaning that the user does not need to know where the data resides.

3.3.6. Workflow and Global Data Management Services.

Both the Workflow and Global Data Management Services are oriented towards managing logistics data (e.g., parts inventory, work in process). Note that Data Management Services and Global Data Management Services are part of the SHAred Data Environment (SHADE) employed by the Department of Defense "Defense Information Infrastructure Common Operating Environment" (DII COE). Concepts of SHADE allow CG data to be shared with external organizations.

3.3.7. Data Interchange Services.

The Data Interchange Service establishes data formats for the interchange of documents, graphics data, and product description data. This promotes the interconnection of applications with database systems within heterogeneous environments, with emphasis on an SQL server interface. This also includes a series of APIs and formats that are needed for portability.

3.3.8. Electronic Data Interchange (EDI) Component.

Electronic Data Interchange provides a mechanism for the electronic exchange of data that would be traditionally conveyed on paper documents. The expected benefits include reduced paperwork, fewer transcription errors, faster response time for procurement and customer needs, reduced inventory requirements and more timely payment of vendors. Electronic exchange of data using EDI is governed according to established rules and formats. The data that are associated with each type of functional document, such as a purchase order or invoice, are transmitted together as an electronic message. The formatted data may be transmitted between originator and recipient via telecommunications or physically transported on electronic storage media. Implementation of EDI requires a family of interrelated standards. The family includes types of messages (also called transaction sets), transmission envelopes, data elements, and short sequences of data elements called data segments.

3.4. Common Support Applications Layer

Unlike Infrastructure Services, Common Support Applications tend to be much more specific to a particular mission domain providing functionality such as presentation graphics and multi- media support. Section 7.1 describes the Coast Guard's list of common software products. These products represent the defacto standard for functional elements of the USCG COE Common Support Applications Layer.

3.4.1. Messaging Processing Services.

Message Processing Services provide the functionality required for preparing, receiving, analyzing, and validating messages. They are also able to normalize message data into formats usable by other applications. Messaging Processing Services provide a user interface to access message handling functions and built-in journalizing functions. Messages can be interactively created and edited in a distributed environment. The Messaging Processing Services analysis capabilities support the filtering of message traffic and assignment to separate windows based on parameters extracted from the message such as recipient, time stamp (date- time group), precedence, and classification. Messages may be generated automatically from operational databases.

3.4.2. Office Automation Services.

The Office Automation Services functional element includes word processing, spreadsheet, briefing support, electronic mail, World- Wide-Web browsers, and other related functions. Web browsers are in the Common Support Applications layer, but Web Servers fall within the Infrastructure Services layer.

3.4.3. Data (Logistics) Analysis.

The Data Analysis functional element contains common functions, such as Pert charts, for analysis and display of logistics-related information.

3.4.4. Developer's Toolkit.

CG COE Services provide the functionality for the development of applications. This includes applications development tools (e.g., ORACLE Developer/Designer/2000, Visual Basic, PowerBuilder, etc.).

3.4.5. MCG&I Mapping Cartography Geodesy & Imagery (MCG&I).

The MCG&I Services functional element processes and displays *National Imagery & Mapping Agency* (NIMA) and *National Oceanographic & Atmospheric Administration* (NOAA) digital maps or other products. The MCG&I Services functional element also process and displays imagery received from various sources, including deployed assets, intelligence command and command centers. An example of an application utilizing this service is the Electronic Chart and Display Information System (ECDIS).

3.4.6. Alerts Service.

The Alerts Service functional element is responsible for routing and managing alert messages throughout the system whether the alert is an "out of paper" message to a systems administrator, a logoff request from an administrator to users, or a "Spill" alert to a watch operator. The Alerts Service builds on the signal processing functions in the Kernel (see Section 3.1) to manage tactical processing on a workstation or, with an alerts server, on a family of workstations. The Alerts Service defines alert and event message formats and controls the posting, reviewing, queuing, and de-queuing of alerts and events. Events can be generated by a process or generated by a system event. The Alerts Service provides:

Alert Generation - Provides a function call that instructs the alerts server to generate a specific alert.

Alert Definition - Provides an interactive interface to create and name routing definitions that include addressing by role, duty group, or user ID.

3.4.7. Correlation Service.

The Correlation Service is responsible for maintaining a consistent view of the *Common Operational Picture* (COP) by correlating information from sensors or other sources that indicate the disposition of platforms of interest.

3.4.8. On-line Help Services.

The Online Help Services functional element provides applications with a uniform technique for displaying context-sensitive help.

3.4.9. Data Access Services.

The Data Access Services functional element is part of SHADE and provides applications with common data-access methods procedures, and tools.

3.5. Applications Programmer Interfaces (API)

An API is the interface between an enterprise application and the underlying platform services. An API specifies the mapping between program syntax and the features of a specific service, and thereby provides access to that service from applications written in a particular programming language, when the application is bound to the service by the language implementation. Standards- based API calls provide a level of application portability and independence from the underlying platform services.

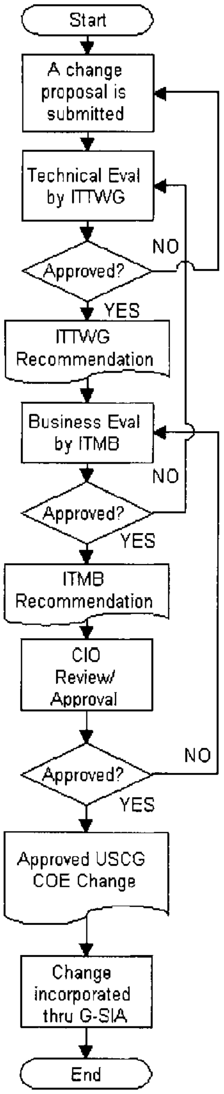
Developers use APIs to request the services of the common system components such as directories, file transfers, E-mail, remote database access, etc. APIs are the means of accessing the service elements of all CG network and platform services. Standards-based APIs will be documented through the USCG COE management process and incorporated in a future revision of the COE.

4. CG COE MAINTENANCE PROCESS

The USCG COE is a living model, subject to periodic update. As new requirements emerge, changes to the U.S. Coast Guard's COE must occur. A well structured configuration management process is required to incorporate these updates. The below graphic depicts the U.S. Coast Guard's COE configuration management process:

G-SIA conducts the USCG COE configuration management process through the Information Technology Technical Working Group (ITTWG), which it chairs. It is through this process the standards for such products are established. The USCG COE should be referenced whenever acquiring IT, to ensure the acquisition complies with the standard products defined. The following flow chart illustrates the process:

U.S. Coast Guard Common Operating Environment (USCG COE)



- A technical architecture change proposal is submitted to Commandant (G-SIA) for coordination. Sufficient resources must be identified to implement and sustain changes to the USCG COE.
- G-SIA processes the proposal through the Information Technology Technical Working Group (ITTWG), who conducts a technical evaluation.
- Upon approval, the ITTWG forwards its findings and recommendations to the ITMB, who performs a business evaluation.
- Upon approval, the ITMB forwards its findings and recommendations to the CIO for approval.
- Upon approval, the change is incorporated into the COE.

The CIO is the sole authority for approving any changes to the USCG COE. The latest version of the USCG COE, a change queue and status, is available from the USCG COE INTRANET web site at:

http://cgweb.uscg.mil/g-s/g-si/g-sia/ita/cgcoe/index.htm.

U.S. Coast Guard Common Operating Environment (USCG COE)

The USCG COE is modeled after The Defense Information Infrastructure Common Operating Environment (DII COE). DISA maintains its DII COE software in an online configuration management repository called SDMS (Software Distribution Management System). This approach decreases the development cycle by allowing developers to receive software updates, or to submit new software functional elements, electronically. With appropriate security measures, installation costs are also reduced because operational platforms may be updated electronically across SIPRNET or other networks. As the USCG COE matures, approaches such as this will be considered for USCG COE maintenance and distribution.

5. IT MANAGEMENT STRATEGY

The Information Technology Management Strategy, COMDTINST 5230.58 (series), provides overall direction for managing Coast Guard Information Technology (IT) resources. The directive establishes the Coast Guard's IT management vision, strategies for achieving the IT management vision, objectives to support the strategies, and specific initiatives to accomplish the objectives. The following concepts expand in further detail, on the Coast Guard IT Management strategies, and the resulting benefits of applying these strategies in conjunction with USCG COE compliance.

5.1. Concepts

All personnel involved with the management of Coast Guard systems shall apply the following concepts throughout the entire system life cycle to realize the benefits of USCG COE compliance described in this document:

- Data sharing - All systems shall be developed in accordance with Coast Guard Data Element Naming Standards, COMDTINST 5230.42 (series). Storing data in common, standard file formats (documents, spreadsheets, databases, images, etc.) using standard data element naming conventions will allow data to be more easily identified, imported from, exported to, and accessed by any application, regardless of location. This will provide more timely, accurate exchange of information for decision making.
- Interoperability - All systems shall be designed in accordance with the Information Systems Technical Architecture, COMDTINST 5230.45 (series) to ensure interoperability with other systems. This includes the ability of all CG systems to share information with other CG systems as well as systems of other organizations, where necessary, to carry out CG missions.
- Reusability - All systems shall employ standard common software to provide common functionality over a common infrastructure. System modules providing common functionality shall be acquired or developed once and used by all systems requiring that common functionality. This practice will reduce acquisition and development time and save development, support, and training costs. This subject is addressed in the Standard Workstation III Configuration Management Policy, COMDTINST 5200.16 (series), as well as the Coast Guard Software Development and Documentation Standards, (CG SDDS), COMDTINST 5234.4 (series).
- Scalability - All systems shall be designed with maximum scalability.

- Portability - All systems shall be designed to allow maximum portability between software operating system environments. All system resource interfaces to the target environment shall be designed to be configurable to the maximum extent possible.

These concepts will improve the quality, and reduce the cost and elapsed time of the following life cycle phases of CG Enterprise

Applications:

- Acquisition - Contract vehicles providing broad access to Commercial Off The Shelf/Government Off The Shelf (COTS/GOTS) products need only be competed once through flexible, open, procurements. Common functionality modules only need to be procured once. Systems are portable avoiding the need to convert when migrating to a new platform. Data is shared so a module to access particular information need only be acquired or developed once.
- Development - COE components only need to be developed once. Those components are shared with future applications leaving one less module to develop. A standard toolset is used to develop systems. Lessons learned and best practices are accrued.
- Deployment - Application configurations are standardized. The target environment configuration is known. Less testing and configuration is required. Applications are installed, they work, and they co-exist peacefully without affecting other applications.
- Training - All applications have a similar look and feel. Each application uses the same keystrokes to do the same thing. All applications become more intuitive, requiring less instruction to operate.
- Operation and Support - As the Coast Guard moves towards a centralized, web-enabled application architecture, applications are fielded from a centralized site. Data resides in a standard Relational Database Management System (RDBMS) installed on a standard server platform running a standard operating system. Users access the application over the Coast Guard Data Network (CGDN+) with a common browser running on a standard desktop workstation. This requires less time to trouble shoot problems. Standard procedures are developed for backup, recovery and contingency for all applications. A standard configuration is maintained throughout the entire Information Architecture.
- Maintaining - The USCG COE provides a well-defined baseline environment. Standard tools are defined, to be used to design, develop, document, test, and monitor change. Wherever possible, a given set of functionality is developed once, then reused in all other applications requiring that same set of functionality. When that module is revised, or enhanced, the same one module is then available to all applications employing it. The time and cost to update each application is avoided.

U.S. Coast Guard Common Operating Environment (USCG COE)

Compliance with the USCG COE will, at a minimum, promote the following goals in support of Coast Guard missions:

- Promote the development and operation of government-wide, inter-operable shared information resources to support performance of missions;
- Reduce fraud, waste, abuse and errors resulting from lack of, or poor implementation of systems;
- Maintain established baselines of timelines and cost parameters for delivery of IT through strategic deployment and implementation of products;
- Reduce end-user training requirements;
- Develop a well-trained core of government IT personnel;
- Reduce end-user and product support requirements;
- Ensure the linkage between investments and Coast Guard missions is maintained.

6. IT IMPLEMENTATION STRATEGIES

Over the last several years, several legislative mandates have been imposed on federal government information system planning activities. These mandates are intended to improve productivity and reduce life cycle costs through improved interoperability and scalability of federal government information systems. Coast Guard IT planning activities evaluate emerging technologies to identify emerging trends which may be leveraged to respond to these mandates. From resulting findings, implementation strategies are developed to best take advantage of these trends. Some examples include centralized data centers, standard database environment guidelines, Electronic Commerce (EC), Electronic Data Interchange (EDI), Digital Signature Standard (DSS), web-enabled centralized application architectures, leveraging COTS/GOTS applications, and the establishment of end-user classes.

These implementation strategies are described in the IT Management Strategy, COMDINST 5230.58, the Coast Guard's annual Information Systems Plan published by Commandant G-SIA, as well as reference (i).

7. STANDARDS, OVERVIEW

"Standards are documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure materials, products, processes and services are fit for their purpose.

For example, the format of credit cards, phone cards, and "smart" cards that have become commonplace is derived from an International Organization for Standards (ISO) international standard. Adhering to the standard, which defines such features as an optimal thickness (0.76 mm), means that the cards can be used worldwide.

International standards contribute to making life simpler and to increasing the reliability and effectiveness of the goods and services we use."

*International Organization for Standards
(ISO), web site
WWW.iso.ch/infoe/intro.htm*

The USCG COE Working Group recognized that IT standards form the basis of the USCG COE. The Information Systems Technical Architecture (ISTA), COMDTINST 5230.45 (series), defines the IT standards of the USCG COE. Each standard described is linked to a respective object within the USCG COE. The standards described in the ISTA must be updated periodically as a result of the rapid changes in the industry. Commandant (G-SIA) coordinates ISTA updates through the USCG COE maintenance process.

7.1. Common Products List

The USCG COE Common Products List describes the current components of the Coast Guard's standard workstation image, which have been approved by TISCOM, and comply with the USCG COE. The most current version of this list may by found on the TISCOM INTERNET site at: http://www.tiscom.uscg.mil/isd/ENG1/ConfMgmt/cfgmgthome.htm

7.2. Certified Enterprise Applications

Shrink-wrapped applications are not to be installed on CG standard workstations unless CG-certified installation instructions are included. U.S. Coast Guard Certified Enterprise Applications (EAs) come with tested, and "TISCOM-certified" installation instructions. Program sponsors are instructed to provide these instructions along with the software and a copy of the TISCOM Certification Certificate when distributing an application for deployment. Attempts to install an uncertified COTS (commercial-off-the-shelf) product using the vendor-supplied, shrink wrapped "Setup.exe" (or equivalent) will not install correctly for the Coast Guard architecture. This is because almost every commercial product -- even the ones that say "network compatible" or "network capable" or that claim to support NTXXX -- are still written to the "lowest common denominator," which is a stand-alone PC. COTS application installation routines copy files to local C:\ drive and system directories which conflicts with the U.S. Coast Guard standard workstation architecture. The certification process results in a common directory structure and configuration where these applications reside on an applications server. A list of certified applications, as well as additional information is available from the following web page:
http://www.tiscom.uscg.mil/isd/ENG1/Certification/SWCert.htm

7.3. Workstation/Server Architecture

Workstation/Server Architecture standards describe the Coast Guard standard workstation desktop computing architecture. This includes the standard directory structure and logical drive configuration for standard workstation end-user file storage on workgroup file servers. This information is essential for planning where prospective enterprise applications will be installed. It also provides the information about the locations of operating system files, shared folders, office automation software, etc. For more information please see appendix (3).

U.S. Coast Guard Common Operating Environment (USCG COE)

Appendix 1 - Index of Related Policy Documents

COMDTINST 3090.6, USCG C4I BASELINE ARCHITECTURES
COMDTINST 3090.7, USCG C4I OBJECTIVE ARCHITECTURE AND TRANSITION PLAN
COMDTINST 4130.6, COAST GUARD CONFIGURATION MANAGEMENT
COMDTINST 5230.2, STANDARD WORKSTATION III OPERATING SYSTEM STANDARD
COMDTINST 5230.42A, COAST GUARD DATA ELEMENT NAMING STANDARDS
COMDTINST 5230.45A, INFORMATION SYSTEMS TECHNOLOGY ARCHITECTURE
COMDTINST 5230.49, USE OF THE USCG OPERATIONS SYSTEM CENTER
COMDTINST 5230.51, COAST GUARD MICROCOMPUTER ALLOWANCE POLICY
COMDTINST 5230.55, ACQUIRING MICROCOMPUTER RESOURCES
COMDTINST 5230.58, U.S. COAST GUARD INFORMATION TECHNOLOGY MANAGEMENT STRATEGY
COMDTINST 5234.4, COAST GUARD SOFTWARE DEVELOPMENT AND DOCUMENTATION STANDARDS (CG-SDDS)
COMDTINST 5234.5A, DESKTOP PRODUCTIVITY SOFTWARE STANDARDS
COMDTINST 5200.16 (SW3 Configuration Management) DRAFT
COMDTINST 5230.XXX (SW3 Software Certification Policy) DRAFT
COMDTINST 5230.59 (CG Common Operating Environment) DRAFT
COMDTINST M5500.13A, AUTOMATED INFORMATION SYSTEMS (AIS) SECURITY MANUAL
COMDTINST M550017, STANDARD WORKSTATION SECURITY HANDBOOK
COMDTINST M5520.12, PERSONNEL SECURITY PROGRAM
ALDIST 056/97 lists the SWIII Migration Schedule Priorities
ALDIST 070/97 addressees interim policy and procedures until a CG-wide RA system is developed.
ALDIST 126/97 Licenses and Maintenance for ORACLE Software Products
ALDIST 213/97 provides policy to units regarding Exchange Mail Server.
ALDIST 252/97 is a data call to collect information for SWIII migration.

Appendix 2 - Glossary

Terms:

Common products list

A list of software products perceived as the products of choice for the functional element they are classified under.

Data Sharing

The practice of storing data in a fashion which allows accessibility by all CG Mission Applications requiring access. Considerations include network/server accessibility and data format.

Information

A framework of guidance for the development, integration, Architecture (IA) deployment, operation, and management of information technology (i.e., computers, communications, and electronics).

Interoperability

The ability of two or more systems to use and exchange information.

Life cycle phases

Steps of activity within the life cycle of a computer application including specification, acquisition, development, deployment, use, and maintenance.

Mission Application

An information system that supports a Coast Guard mission.

OA&TP

The Objective Architecture and Transition Plan (OA&TP) represents the Coast Guard's first organizational master plan for collecting, processing, and exchanging information by which decision makers from operations and mission support programs, systems support, budgeting and acquisition offices, plan for attaining Coast Guard Command, Control, Communication, Computers, Intelligence and Sensor capabilities."

U.S. Coast Guard C4I Objective Architecture and Transition Plan, Executive Summary

Portability

The ease with which a system or component can be transferred from one hardware or software environment to another with minimal change. This allows hardware upgrades to refresh technology with minimal impact on resident applications.

Re-usability

The concept of reusing generic modules of computer code to provide the same function to all applications requiring that function.

Scalability

The ability to for the same software application to operate on any platform from a personal computer to a super computer, depending on user requirements.

U.S. Coast Guard Common Operating Environment (USCG COE)

Appendix 3 - Workstation/Server Architecture

Overview: The following information represents some general concepts behind the Coast Guard's NT Workstation/Server Architecture implementation, which should help developers "target" their applications for the U.S. Coast Guard standard workstation environment.

Application Code Location: In the U.S. Coast Guard standard workstation client-server environment what goes on the client is, by and large, already there. Items that go on the desktop (laptop) workstation include:

- the operating system
- MS Office suite
- Mail Client (Exchange/Outlook)
- Dr. Solomon's anti-virus toolkit and WinGuard
- Internet Browser

With few exceptions, nothing else is allowed on the client:
Exceptions include .DLL or other similar types of files that, for whatever reason, **cannot** work correctly from the Server; registry settings for specific applications; initialization or configuration files specific to an application.
NO executables are allowed on the workstation.
NO permanent data files are allowed on the workstation.
TEMPORARY "scratch" space for applications, to which users have full read/write/delete access, is provided in the C:\TEMP directory. This is **temporary** file space and is deleted upon each logon/logoff of the user.

All users have several default network drives mapped during user logon by virtue of a **logon.cmd** script. These drive/directory locations are:
H:\ -- the Apps share on the Application Server.
O:\ -- the NetApps share on the Application Server.
U:\ -- the user's personal file share on their specific File Server.

The H:\ and U:\ drive mappings are part of the original NT3.51 system configuration/architecture. The O:\ drive mapping is a new addition due to NT4.0 and Zero Admin Kit considerations.

Applications should be written to a specific (developer-defined, TISCOM approved) subdirectory under either the Apps or NetApps share on the Application Server.

ORACLE code location The current SWIII configuration has the runtime files for ORACLE Forms, Reports and both SQL*Net and Net8.
These are located in the BIN subdirectory of the Apps share (H:\ drive).

ACCESS code location As previously stated, MSOffice is loaded to the client, C:\MSOffice\... Access is located there.

INFORMIX interpreter code location There is no INFORMIX loaded as part of the default SWIII standard configuration.

Data file locations Depends upon the type of data to be stored:

- Data that all users need to SHARE for a specific application: set storage for a developer-determined subdirectory under an appropriate application subdirectory in the **Apps** share. (E.g. D:\NSAPPS\MyApplication\MyData).
- Data that is of a temporary nature during program execution: utilize C:\TEMP on the client.
- Data that is unique for each application user: set up a specific location (if required so that it can be "hard coded" into an application) or a user-specified location under the user's U:\ drive (personal file storage) (e.g. U:\AppName\AppData).

Server Space Limitations.

The original SWIII architecture only provided 2-3Gbytes of space on the Application Server. This space was intended/purchased for the 19 or 20 Mission Essential Applications designated to be ported from CTOS to SWIII. There has been no clear policy set yet on what happens when a new application comes along that fills the hard drive, or, more likely, when an MEA is deployed which requires more space than is available, and non-MEAs are loaded on the Apps Server.

Standard Workstation recommended design practices

- NOTHING goes on the client (save for the exceptions noted previously).
- Every application SHOULD run from the Apps Server.
- Any unique files, .DLLs that are "updates" or "backdates" of existing system .DLLs, and all data for an application should reside in an application-specific directory under the **Apps** or **NetApps** file share.
- In the NT4.0 implementation items in the **NetApps** share can be made "visible" to all users without having to field Icons to each individual desktop. For that reason **NetApps** is the preferred location for new applications.